

CSCI 4302/5302: Advanced Robotics

Homework 1: Due Monday, September 13, 11:59 p.m. to Canvas

Learning objective: Cultivate basic skills with ROS, the Robot Operating System.

ROS, the Robot Operating System, serves as middleware and a convenient hardware abstraction layer when working with mobile platforms, sensors, and algorithms. It was originally built to run on top of a BSD/Linux kernel in user space. (You will have the easiest time installing ROS on the Ubuntu 20.04 LTS release.)

The current recommended configuration is running **ROS Noetic** (ROS1, not ROS2) on Windows 10 or Ubuntu 20.04 LTS operating system. Mac OS installation instructions are below, but should be considered experimental and we will not be able to offer much in the way of system technical support for issues stemming from non-Ubuntu installations should you prefer to use an alternate configuration.

Your assignment for this week and next is to complete the following **Tutorials**:

- | | |
|--|---|
| 1. Installing and Configuring Your ROS Environment. | 8. Writing a Simple Publisher and Subscriber (Python) |
| 2. Navigating the ROS Filesystem | 9. Examining the Simple Publisher and Subscriber |
| 3. Creating a ROS Package | 10. Understanding ROS Services and Parameters |
| 4. Building a ROS Package | 11. Creating a ROS msg and a ROS srv |
| 5. Understanding ROS Nodes | 12. Writing a Simple Service and Client (Python) |
| 6. Using <code>rqt_console</code> and <code>roslaunch</code> (2.2 onward only) | 13. Examining the Simple Service and Client |
| 7. Understanding ROS Topics | |

What you turn in to Canvas:

- Create a Catkin package named “<identikey>-hw1” replacing <identikey> with your CU Identkey.
- Write and test a publisher node that publishes a *topic* and a subscriber node that subscribes to *topic*. Measure the time it takes for the subscriber to receive messages over *topic* once they are published (Hint: You can set/use the timestamp field in the message header) and generate a histogram of this latency with matplotlib for 300 messages.
- Write and test a server node that provides a *service* and client node that calls this *service*. This service should take a string as input, and return the same string as output. Over 300 service calls, measure the length of time it takes for a call to execute and return to the client and generate a histogram of these arrival times.
- Upload an image or document showing the histograms of latencies to Canvas.
- Upload a zip file containing your Catkin package

Resources:

- [Matplotlib Histogram Reference](#)
- [ROS Installation Documentation](#)
- [ROS Standard Messages](#)
- [Mac OS ROS Install Guide](#)